CLAIMS

1. A method for pipelining a table function in a database system, comprising:

a) performing a setup operation when a table function is a called;

b) fetching a subset of output data from a data producer;

c) sending the subset of the output data to a first consumer of the output data, wherein the first consumer is the table function;

d) repeating steps b) and c) until all the output data has been fetched from the data producer.

2. The method of claim 1 in which the act of performing a setup operation comprises setting up a context object to maintain state.

3. The method of claim 1 in which the data producer comprises a second table function.

4. The method of claim 1 in which the subset of the output data comprises a single data object or row of data.

5. The method of claim 1 in which the subset of the output data comprises a plurality of data objects or rows of data.

6. The method of claim 1 further comprising:

e) performing a close operation after all the output data has been fetched from the data producer.

7. The method of claim 6 in which the act of performing the close operation comprises garbage collection operations.

8. The method of claim 7 in which the garbage collection operations comprises removal of a context object.

9. The method of claim 1 in which the table function executes in a different execution thread than the data producer.

10. The method of claim 1 in which the table function and the data producer execute from an identical execution thread.

11. The method of claim 1 in which a callback function is passed from the table function.

12. The method of claim 11 in which the callback function is executed on each subset of the output data fetched from the data producer.

13. The method of claim 1 in which the data producer comprises a dynamically configurable return type.

14. The method of claim 13 in which the dynamically configurable return type is established at compile time.

15. The method of claim 1 in which steps a) through d) are implemented within a database query language statement.

16. The method of claim 15 in which the database query language statement comprises SQL.

17. The method of claim 1 in which the subset of the output data is pipelined to a database query language statement.

18. The method of claim 17 in which a callback function is invoked for the subset of the output data.

19. The method of claim 18 in which the callback function filters inappropriate data.

20. The method of claim 1 further comprising:

e) send the subset of the output data to a second consumer of the output data.

21. The method of claim 20 further comprising the step of determining whether the subset of the output data should be routed to the first consumer or the second consumer;

executing step c) if the subset of the output data should be routed to the first consumer; and

executing step e) if the subset of the output data should be routed to the second consumer.

22. The method of claim 21 in which a partitioning definition is applied to determine whether the subset of the output data should be routed to the first consumer or the second consumer.

23. The method of claim 22 in which the partitioning definition comprises either hash or range based partitioning.

24. The method of claim 1 in which the first consumer processes the subset of the output data in parallel.

25. The method of claim 24 in which multiple slaves exist to process the subset of the output data.

26. The method of claim 25 further comprising the step of determining which of the multiple slaves operate upon the subset of the output data.

27. The method of claim 26 in which a partitioning definition is established to route the subset of the output data to an appropriate one of the multiple slaves.

28. The method of claim 27 in which the partitioning definitions comprises either hash or range based partitioning.

29. The method of claim 1 further comprising:

optimizing a query comprising the table function.

30. The method of claim 29 in which statistics for the table function are passed to an optimizer.

31. The method of claim 29 in which an optimizer self-determines statistics to optimize the query.

32. A system for pipelining table functions, comprising:

a) means for performing a setup operation when a table function is a called;

b) means for fetching a subset of output data from the table function;

c) means for sending the subset of the output data to a first consumer of the output data;

d) means for repeating steps b) and c) until all the output data has been fetched from the table function.

33. A computer program product comprising a computer usable medium having executable code to execute a process for pipelining table functions, the process comprising the steps of:

a) performing a setup operation when a table function is a called;

b) fetching a subset of output data from the table function;

c) sending the subset of the output data to a first consumer of the output data;

d) repeating steps b) and c) until all the output data has been fetched from the table function.